CLAIMS:

1. A compile method in a compiler suitable for a speculation mechanism, wherein said compiler generates object codes for a processor having a speculative instruction and a speculative check instruction for checking a speculation failure (said speculative instruction and speculative check instruction are generally called a "speculation mechanism"), said compile method comprising the steps of:

(a) generating first object codes using said speculation mechanism from a repetitively executed fragment of a source program;

(b) generating second object codes not using said speculation mechanism from said repetitively executed fragment of said source program; and

(c) generating third object codes that perform a control transfer so that after a number of times a speculation failure is detected by said speculative check instruction during execution of said first object codes satisfies a predetermined condition, said second object codes for said repetitively executed program fragment are executed.

2. A compile method suitable for a speculation mechanism according to claim 1, wherein said predetermined condition in said step (c) is that the number of times a speculation failure is detected exceeds a predetermined value.

3. A compile method suitable for a speculation

mechanism according to claim 1, wherein said predetermined condition in said step (c) is that a ratio of the number of times a speculation failure is detected by the speculation check to a number of times the repetitively executed program fragment is executed exceeds a predetermined value.

4. A compile method suitable for a speculation mechanism according to claim 1, wherein when a speculation failure is detected by the speculation check, a value of counter is incremented and when the counter value exceeds a predetermined value, said third object codes transfer control to execution of said second object codes.

5. A compile method suitable for a speculation mechanism according to claim 1, wherein once said speculation failure is detected, said third object codes transfer control to execution of said second object codes.

6. A compiler program using said compile method according to claim 1.

7. A storage medium storing the compiler program according to claim 6.

8. A compile method for generating an object program from a source program including repetitive loop processing, said compile method comprising the steps of:

generating first object codes from said source program by using a speculative instruction and a speculative check instruction for checking a speculation failure;

generating second object codes from said source program without using said speculative instruction and said speculative check instruction;

generating third object codes that perform control to first execute said first object codes;

generating fourth object codes to count a number a times the speculation failure occurs during execution of said first object codes; and

generating fifth object codes that perform control to execute said second object codes after the number of times reaches a predetermined value.

9. A computer for generating an object program from a source program including repetitive loop processing, comprising:

a memory device to store said source program;

a central processing unit (CPU) to execute a compiler program for generating said object program from said source program;

a display device to output a result of compile processing executed by said CPU; and

a bus to connect said memory device, said CPU and said display device;

wherein said CPU generates said object program by executing a compiler program that includes the steps of:

generating first object codes from said

source program by using a speculative instruction and a speculative check instruction for checking a speculation failure;

generating second object codes from said source program without using said speculative instruction and said speculative check instruction;

generating third object codes that perform control to first execute said first object codes;

generating fourth object codes to count a number of times a speculation failure occurs during execution of said first object codes; and

generating fifth object codes that perform control to execute said second object codes after the number of times reaches a predetermined value.

10. An object program generated from a source program including repetitive loop processing including:

a first object code portion generated from said source program by using a speculative instruction and a speculative check instruction for checking a speculation failure;

a second object code portion generated from said source program without using said speculative instruction and said speculative check instruction;

a third object code portion that performs control to first execute said first object code portion;

a fourth object code portion to count a number of times a speculation failure occurs during execution of said first object code portion; and

a fifth object code portion that performs control to execute said second object code portion after the number of times reaches a predetermined value.

11. A storage medium storing the object program according to claim 10.